

UNITED STATES PATENT APPLICATION

For

INPUT DATA SELECTION FOR CONTENT ADDRESSABLE MEMORY

INVENTOR:

Varadarajan Srinivasan

Prepared By:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP  
12400 WILSHIRE BOULEVARD  
SEVENTH FLOOR  
Los Angeles, CA 90025-1026  
(408) 720-8300

"Express Mail" mailing label number: EL 351960985W  
Date of Deposit: 1/14/02  
I hereby certify that I am causing this paper or fee to be deposited with the  
United States Postal Service "Express Mail Post Office to Addressee" under 37  
C.F.R. § 1.10 on the date indicated above and that this paper or fee has been  
addressed to the Assistant Commissioner for Patents, Washington, D. C. 20231  
JUANITA KRISCOE  
(Typed or printed name of person mailing paper or fee)  
[Signature]  
(Signature of person mailing paper or fee)  
1/14/02  
(Date signed)

2004754-4524007

## INPUT DATA SELECTION FOR CONTENT ADDRESSABLE MEMORY

### FIELD OF THE INVENTION

[0001] This invention relates generally to the field of memory devices. More specifically, the invention relates to content addressable memory devices.

### BACKGROUND OF THE INVENTION

[0002] Networks contain a collection of computing systems (e.g., clients and servers) that are interconnected by transmission lines to enable the transfer of data between them.

A network typically includes multiple access points (e.g., routers and servers) that may switch and/or route data between transmission lines to transfer data from a source to a destination. Data is typically transmitted in the form of packets that are made up of smaller data cells. A packet is a unit of data that is routed between a source and a destination on a packet-switched network. When a file (e.g., e-mail, graphics, etc.) is sent from one place to another on a network, the file is divided into such smaller packets making them more efficient for transmission. The individual packets for a given file may travel different routes throughout networks with each packet containing both data and transmission information associated with the routing of data. As such, a packet may be described as having a payload containing the data, and one or more headers that contain the routing information (e.g., a destination address).

[0003] When all the packets have arrived at a destination, they are reassembled into the original file at the receiving end. Such a packet switching scheme is an efficient way to handle transmission on a connectionless network. This is in contrast to a circuit switching scheme where a connection (e.g., a voice connection) requires the dedication of a particular path for the duration of the connection.

2004754-01402

[0004] A router is a device (e.g., hardware, firmware, software) that determines the next network segment to which a packet should be forwarded towards its destination. A router may be positioned at points within a network or where one network meets another, referred to as a gateway. A router may create and maintain tables of the available routes and their conditions for use with other information to determine the best route for a given packet. Typically, a packet may travel through a number of network points having routers before arriving at its destination.

[0005] When a data packet arrives at the input of a router, several lookups may be performed to determine the subsequent handling of the packet, as illustrated in Figure 1. The lookups may include, for examples, where to send the packet next (Next Hop), the quality of service requirement (QoS), the Ethernet port address, etc. Consider, for example, a packet arriving at Router-A. Router-A needs to determine whether the packet is destined for local servers connected directly to Router-A, or if the packet should go to the next router on a route (Router-B) to a destination. Additionally, Router-A may assign a priority based on the destination address (DA) and the source address (SA) of the packet.

[0006] The packet header may first be parsed or processed to get the values from different fields (e.g., SA, DA, protocol type, QoS, etc) in order to perform the various lookups. A packet classification lookup, for example, may be performed using SA, DA and other relevant fields in the packet header. The Next Hop lookup, for example, may also be performed to determine whether the packet is meant for local servers or for Router-B. If the packet is destined for Router-B, the packet is then put in a queue for Router-B. If the packet is destined for a local server (e.g., Server-1 or Server-2), then a

media access control (MAC) lookup is performed to send the packet to the appropriate server. In the preceding example, three lookups are necessary for sending the packet on its way: Packet Classification, Next Hop, and MAC. However, often there are other lookups performed on the packet header, with the number of lookups exceeding five or more.

[0007] Routers may use processors and content addressable memory (CAM) devices to perform the various lookups on packets. The CAM device can be instructed by a processor to compare a search key, also referred to as comparand data (e.g., packet header data), with data stored in its associative memory array, as illustrated in Figure 2. The CAM simultaneously examines all of its entries and selects the stored data that matches the key.

[0008] When the entire CAM device, or blocks thereof, is searched simultaneously for a match of the stored data with the key comparand data, the CAM device indicates the existence of a match by asserting a match flag. Multiple matches may also be indicated by asserting a multiple match flag. The CAM device typically includes a priority encoder to translate the matched location into a match address or CAM index and outputs this address to a status register so that the matched data may be accessed. The priority encoder may also sort out which matching memory location has the top priority if there is more than one matching entry.

[0009] Data may be represented in the form of strings of binary digits ("bits") having a low ("0") logic state and a high ("1") logic state. Different types of CAMs may be used with different data formats. A binary CAM is designed to operate with "0" and "1" states, while a ternary CAM is designed to operate with "0", "1", and "don't care" states.

The bits may be organized into groups such as a word (e.g., 64 or 72 bits wide) and stored in different segments of a CAM. The keys used for different data fields may have different word sizes, for example, the key for a Classification lookup may be 128 bits wide and the key for a Next Hop lookup may be 32 bits wide.

[0010] A router may include multiple CAMs, with each CAM having a different table or, alternatively, a single CAM having multiple blocks for each of the different tables, for performing the different lookups. For example, a router may include a 32 bit wide Next Hop CAM, a 128 bit Classification CAM, and a 48 bit MAC CAM. With routers having multiple CAMs, each of the multiple CAMs are typically connected to common buses that are used to communicate the various keys and other input and output data with each of the CAM devices. Similarly, with routers having a single CAM with multiple blocks, each of the blocks is accessed using common buses. Thus, lookups are typically performed sequentially before a packet is processed (e.g., routed to the next destination or classified). Because the buses are shared with so many input and output functions of all the CAMs or CAM blocks, many clock cycles are required to multiplex data on the bus. This generally limits the search rate and overall throughput of conventional CAM devices. As the number of ports, segments, or devices that are supported by routers and as the number of lookups increase, conventional CAM devices and architectures can undesirably limit the system's overall throughput.

## BRIEF DESCRIPTION OF THE DRAWINGS

[0011] The present invention is illustrated by way of example and not intended to be limited by the figures of the accompanying drawings.

[0012] Figure 1 illustrates an example of packet handling by a router.

[0013] Figure 2 illustrates one embodiment of a conventional CAM device.

[0014] Figure 3 illustrates one embodiment of a line card or blade of a router having a CAM device configured according to one embodiment of the present invention.

[0015] Figure 4 illustrates one embodiment of a lookup string.

[0016] Figure 5 illustrates one embodiment of a CAM device.

[0017] Figure 6 illustrates one embodiment of the filter circuit of Figure 5.

[0018] Figure 7 illustrates one embodiment of the select logic of Figure 6.

[0019] Figure 8 illustrates one embodiment of a comparand register.

[0020] Figure 9 illustrates one embodiment of a cross-bar switch.

[0021] Figure 10 illustrates one example of an input string filtered into a comparand register.

[0022] Figure 11 illustrates one example of a programmed cross-bar switch.

[0023] Figure 12 illustrates one embodiment of programmed select logic.

[0024] Figure 13 illustrates one example of an input string filtered into a comparand register.

[0025] Figure 14 illustrates one example of a programmed cross-bar switch.

[0026] Figure 15 illustrates another embodiment of the select logic of Figure 5.

204754-0140

[0027] Figure 16 illustrates one embodiment of a CAM cell pair for the select logic of Figure 15.

[0028] Figure 17 illustrates another embodiment of a CAM device.

2017-07-10 15:40:00

## DETAILED DESCRIPTION

[0029] In the following description, numerous specific details are set forth such as examples of specific, components, circuits, processes, etc. in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that these specific details need not be employed to practice the present invention. In other instances, well known components or methods have not been described in detail in order to avoid unnecessarily obscuring the present invention.

[0030] Embodiments of the present invention include various method steps, which will be described below. The steps may be performed by hardware components or may be embodied in machine-executable instructions, which may be used to cause hardware components (e.g., a processor, programming circuit) programmed with the instructions to perform the steps. Alternatively, the steps may be performed by a combination of hardware and software.

[0031] Embodiments of the present invention may be provided as a computer program product, or software, that may include a machine-readable medium having stored thereon instructions. The machine readable medium may be used to program a computer system (or other electronic devices) to generate articles (e.g., wafer masks) used to manufacture embodiments of the present invention. The machine-readable medium may include, but is not limited to, floppy diskettes, optical disks, CD-ROMs, and magneto-optical disks, ROMs, RAMs, EPROMs, EEPROMs, magnet or optical cards, flash memory, or other type of media / machine-readable medium suitable for storing electronic instructions.



[0032] The machine readable medium may store data representing an integrated circuit design layout that includes embodiments of the present invention. The design layout for the integrated circuit die may be generated using various means, for examples, schematics, text files, gate-level netlists, hardware description languages, layout files, etc. The design layout may be converted into mask layers for fabrication of wafers containing one or more integrated circuit dies. The integrated circuit dies may then be assembled into packaged components. Design layout, mask layer generation, and the fabrication and packaging of integrated circuit dies are known in the art; accordingly, a detailed discussion is not provided.

[0033] The methods and apparatus described herein may be used to translate input data (e.g., an input string) to one or more comparands for searching in a content addressable memory. In one embodiment, an apparatus includes one or more filter circuit that filter (e.g., compacts, duplicates and/or transposes) the input data, according to the needs of the user. The filter circuit includes a switch circuit (e.g., a cross-bar switch, switch matrix, switch fabric, etc.) and select logic. The select logic selects or enables particular programmed switch paths of the switch circuit to provide one or more bits or groups of bits from the input data to particular positions in the comparand string. For one embodiment, the input data has a larger number of bits than the input data bus width of the CAM device such that the entire input data is loaded into the CAM device over the input data bus in several load operations (e.g., over several clock cycles). The select logic may select or enable particular programmed switch paths of the switch circuit in each of the different load operations. The CAM array may also be segmented into several segments of CAM cells with each segment having a corresponding filter circuit.

[0034] Figure 3 illustrates one embodiment of a line card or blade of a router having a CAM device. Line card 300 includes processor 310, ingress interface circuitry 330, egress interface circuitry 340, CAM device 320, associated data storage unit 370, traffic manager 360, and payload storage unit 350.

[0035] Processor 310 functions to control the overall operation of line card 300 in cooperation with the other components of line card 300. For example, processor 310 receives packets from a network medium through ingress interface circuitry 330, stores the payload of packets in payload storage unit 350, and processes packet header information to determine required lookups in CAM device 320 and subsequent handling of the packets, as discussed herein. Ingress circuitry includes, for example, PHY and MAC devices. Processor 310 sends out packets on a network medium through egress interface circuitry 340 based on the lookups performed by CAM device 320. Egress interface circuitry 340 may be connected to a switch fabric or directly to one or more other routers or switches. Processor 310 may be one or more network processor units (NPUs), microprocessors, or one or more special purpose processors such as a digital signal processor (DSP). In another embodiment, processor 310 may be another type of controller, for example, a field programmable gate array or a general purpose processor. The processor 310, ingress interface circuitry 330, and egress interface circuitry 340 components of a router are known in the art; accordingly, a detailed discussion is not provided.

[0036] In response to information in a packet header, for a particular packet, processor 310 determines the number and types of lookups to be performed by one or more of CAM devices 320, and forms the search keys for these lookups. The searches or

lookups may include, for example, classification lookups, forwarding lookups (e.g., Next Hop or longest prefix match (LPM) lookup, MAC lookup, MPLS lookup, etc.). When multiple searches are required, processor 310 forms a composite search key that includes at least two, and as many as all, of the various search keys for the lookups. The composite search key is provided as a common input string to CAM device 320. CAM device 320 selectively identifies and extracts the individual search keys from the common input string and provides the individual search keys to the associated CAM blocks to perform the lookups. Advantageously, the lookups can then occur concurrently or simultaneously in the CAM blocks of CAM device 320, thereby increasing overall throughput over conventional systems in which searches are processed sequentially.

[0037] CAM device 320 may be a multiple block CAM device with each block capable of storing a different table for comparand lookups. Alternatively, CAM device 320 may represent multiple, single block CAM devices (e.g., with each single block CAM device formed on a different integrated circuit substrate) with each CAM device used to store a different table for comparand lookup. After one or more lookups are executed in CAM device 320, associated information for matching entries (e.g., additional routing information and/or packet information) may be retrieved from associated data unit 370. Processor 310 then communicates with traffic manager 360 to schedule the exit of a packet from line card 300 via egress interface circuitry 340.

[0038] Figure 4 illustrates one embodiment of an input string. Input string 400 includes field segments parsed or processed from one or more packet headers 402 and 404. When data processing systems (e.g., routers, clients, servers) exchange data over a network, the procedure involves the use of protocols by which these systems agree on

how to communicate with each other. To reduce design complexity, networks may be organized as a series of layers. The number of layers and the function of each layer varies from network to network.

[0039] For example, where a transmission control protocol (TCP)/Internet protocol (IP) is used, it is organized into multiple layers including a network access layer and an Internet layer. The network access layer uses a TCP to enable the exchange of data between an end system and a network. An Internet layer uses an IP to enable data to transverse multiple interconnected networks. Each of these protocols use packet headers containing routing information, as discussed above. For example, TCP packet header 402 includes a source address (SA) port segment 406 and a destination address (DA) port segment 408, and IP packet header 404 includes a SA segment 410, a DA segment 412, a type of service (ToS) segment 414, and a protocol type segment 416.

[0040] In one embodiment, for example, processor 310 of Figure 3 may be used to parse certain segments from packet headers 402 and 404 to generate input string 400 and transmit the input string to CAM device 320. For example, input string 400 may include MAC segment 418, TOS segment 414, SA port segment 406, DA port segment 408, SA segment 410, and DA segment 412. Alternatively, input string 400 may include more or less than the segments illustrated. A filter circuit (illustrated in Figure 5) may then filter out the bit values of different field segments of input string 400 to generate different filtered comparand strings to concurrently perform different lookups in the CAM blocks. In an alternative embodiment, processor 310 may transmit as-received unparsed header segments to CAM device 320.

[0041] Figure 5 illustrates CAM device 500 that is, for example, one embodiment of CAM device 320 of Figure 3. CAM device 500 includes address circuit 512, instruction decoder 510, CAM array 502, read/write (R/W) circuitry 514, priority encoder 516, match flag logic 518, comparand register 504 and filter circuit 506.

[0042] CAM array 502 is an array of CAM cells that includes any number of rows of L bits of CAM cells that may be any types of CAM cells including, for example, binary and ternary CAM cells. Each CAM cell includes at least one memory storage element and at least one compare circuit. Other embodiments may be used to effectively implement an array of CAM cells.

[0043] CAM array 502 is coupled to comparand register 504 that receives input data from filter circuit 506 for lookup in CAM array 502. Comparand register 504 has an input bus width of L bits, and is coupled to filter circuit 506 that also has an input bus width of L bits. Comparand register 504 CAM device 500 receives input string 508 over comparand bus CBUS also having a width of L bits. The input string includes comparand data for comparison with data stored in one or more CAM cells of CAM array 502. Input string 508 may be generated, for example, by processor 310 of Figure 300. Input string 508 may be, for example, input string 400 of Figure 4. Note that, in an alternative embodiment, the number of bits in a row of CAM cells in CAM array 502 and/or the input bus width of comparand register 504 and/or the input bus width of filter circuit 506 may be different than L. The CAM device 500 may also include one or more global mask registers (not shown) for the comparand data provided to CAM array 502.

[0044] Input string 508 includes M bits, where M is greater than or equal to L. The input string is transferred over the CBUS in N segments, where N is at least M/L. If M is

equal to L then all M bits of input string 508 are delivered in a single segment to filter circuit 506. If M is larger than L, for example, if M is 288 bits and L is 72 bits, then the 288 bits can be transferred over the CBUS in a minimum of four segments ( $4=288/72$ ), with each segment having 72 bits of the input string. In another example, if M is 288 bits and L is 100 bits, then the 288 bits can be transferred over the CBUS in a minimum of three segments ( $2.88=288/100$ ) with, for example, two segments having 100 bits and the last segment having 88 bits.

[0045] Alternatively, one or more of the N segments may have a different number of bits. For example if M is 288 bits and L is 100 bits, then 98 bits, for example, may be transferred over the CBUS in a first segment, 96 bits in a second segment and 94 bits in a third segment.

[0046] Filter circuit 506 receives filter data (FDATA), that is used to directly or indirectly program filter circuit 506 to generate a particular filtered comparand string from input string 508. For example, filter circuit 506 may be a cross-bar switch, switch matrix, or switch fabric that includes one or more programmable circuit elements (e.g., programmable memory storage elements) that can be programmed (e.g., in response to FDATA) to select one or more bits or groups of bits of the input string and provide them to comparand register 504. Filter circuit 506 can output the selected bit(s) to comparand register 504 in the same order as received over the CBUS, or in a different placement order (e.g., filtered out, translated, compacted, transposed, and/or duplicated).

[0047] Filter circuit 506 also receives segment select data (SSDATA) that indicates which of the programmed circuit elements (i.e., programmed in response to FDATA) within filter circuit 506 will be selected at any given time to provide at least one bit from

input string 508 to at least one bit position of comparand register 504. SSDATA is provided, for example, together with each segment of the input string during a compare operation. Filter circuit 506 also receives a clock signal CLK to time one or more operations of filter circuit 506. FDATA and SSDATA may be provided by, for example, by processor 310 of Figure 3. In some instances, SSDATA indicates that none of the programmed circuit elements within filter circuits 506 are selected to output one or bits of the input string to the comparand register.

[0048] Instruction decoder 510 decodes various instructions provided on instruction bus IBUS (e.g., instructions to write or read data from CAM array 502 and to compare comparand data with one or more segments of the CAM array) and provides various control and/or timing signals to address circuit 512, filter circuit 506, R/W circuitry 514, and comparand register 504, to control when CAM device 500 performs an operation (e.g., a read, write or compare operation). Instruction decoder 516 may also provide various control and/or timing signals to CAM array 502, priority encoder 516 and match flag logic 518. The instructions are timed by CLK. In an alternate embodiment, instruction decoder 510 is omitted and various read, write and compare control signals are provided directly to one or more of the circuit blocks.

[0049] Data is written to CAM array 502 by read/write circuitry 514 and address logic 512. Address logic 512 selects one or more rows of CAM cells in response to an address provided over ADDR. The write data is provided to the selected cells (e.g., over one or more data bit lines) by the write portion of read/write circuitry 514 (e.g., write buffers). Data is read from one or more selected rows of CAM cells by the read portion of read/write circuitry 514 (e.g., by one or more sense amplifier circuits).

[0050] When CAM array 502 is searched for a match of the stored data with the key comparand data stored in comparand register 503, each matching location indicates a match on one of match lines 403. Match flag logic 518 indicates the existence of a match if at least one of the match lines 503 carries a match signal indicating a match. Additional flag logic such as almost full flag logic, full flag logic, and/or multiple match flag logic may also be included in CAM device 500.

[0051] Priority encoder logic 516 translates a matched location(s) into an index (or a match address) and outputs this index that may be used, for example, by processor 310 of Figure 3 or to access associated data unit 370 of Figure 3. Priority encoder logic 516 also identifies which matching location has the top priority if there is more than one matching entry.

[0052] Any of the signals provided over IBUS, DBUS, CBUS or ADDR may be time multiplexed with other signals and provided over one or more of the other buses.

[0053] Figure 6 illustrates filter circuit 521 that is one embodiment of filter circuit 506 of Figure 5. Filter circuit 521 includes switch circuitry 520, program circuit 522 and select logic 524. Switch circuitry 520 includes a plurality of switch circuits, or paths (not shown), which are programmed by program circuit 522 to connect the input data bits with positions in comparand register 504. For example, switch circuitry 520 may be a cross-bar switch, switch matrix, or switch fabric that includes one or more programmable circuit elements (e.g., programmable memory storage elements) that can be programmed by program circuit 522 (e.g., by PDATA) to select one or more bits or groups of bits of the input string and provide them to comparand register 504. Switch circuitry 520 can output the selected bit(s) to comparand register 504 in the same order as received over the



CBUS, or in a different placement order (e.g., filtered out, translated, compacted, and/or duplicated).

[0054] Program circuit 522 programs switch circuitry 520 in response to FDATA, an address, and a program control signal PGM provided, for example, by instruction decoder 510 (or alternatively by processor 310 of Figure 3, or any other component). PGM indicates when program circuit 522 is enabled to program switch circuitry 520. The address provided by program circuit 522 identifies one or more rows and/or columns of program switch circuits within switch circuitry 520 that will be programmed by PDATA. Several embodiments of the program circuit 522 is described in co-pending United States Patent Application Serial number 09/935,997, entitled "Concurrent Searching of Different Tables Within a Content Addressable Memory" filed on August 22, 2001, which is incorporated herein by reference in its entirety.

[0055] In an alternate embodiment, program circuit 522 may be omitted and the address and PDATA provided directly to switch circuitry 520 from outside the CAM device incorporating filter circuit 521 (e.g., by processor 310 of Figure 3).

[0056] Select logic 524 is configured to receive SSDATA that indicates which of the programmed circuit elements within switch circuitry 520 will be selected at any given time to provide at least one bit from input string 508 to at least one bit position of comparand register 504 (e.g., if the programming indicates such a connection). During a program operation, particular segments or groups of programmed switch circuits in switch circuitry 520 are programmed by SSDATA such that they are assigned segment identifiers as indicated by segment select signals SEL. SSDATA is programmed into select logic 524 when the program control signal R/W\* is in a low logic state. SSDATA

2004754-01402

is read from select logic 524 when the program control signal R/W\* is in a high logic state. During a compare operation, the control signal CMP is asserted to a high logic state and the SSDATA indicates which segment or groups of programmable switch elements are to be enabled. For another embodiment, CMP may be omitted and select logic 524 always enabled to compare.

[0057] If the SSDATA matches a programmed SSDATA for particular segment, the corresponding SEL signal is enabled to an active high logic state, while other SEL signals are disabled. In some instances, none of the SEL signals will be asserted if the programmed SSDATA does not match the input SSDATA. R/W\* and CMP may be provided by instruction decoder 510 (e.g., in response to read, write or program, and compare instructions) or directly by another component (e.g., by processor 310 of Figure 3). Alternatively, CMP and/or R/W\* may be omitted.

[0058] For an alternative embodiment, SSDATA does not carry information for both the read/program function associated with R/W\* and the compare function associated with CMP; rather, separate signal paths may be used for each function.

[0059] For another embodiment, SSDATA is provided directly to switch circuitry 520 without passing through select logic 524.

[0060] Figure 7 illustrates select logic 531 that is one embodiment of select logic 524 of Figure 6. Select logic 531 includes L compare circuits 526<sub>0</sub>-526<sub>L-1</sub>, each coupled to a corresponding memory circuit 528<sub>0</sub>-528<sub>L-1</sub>. The memory circuits each store one (or more) bits of SSDATA during a program or write operation (i.e., when R/W\* is in a logic low state). Each of the compare circuits compares the contents of the corresponding memory circuit with SSDATA during a compare operation (i.e., when CMP is in a high

logic state). The comparison results are reflected on the SEL signal lines 530<sub>0</sub>-530<sub>L-1</sub>.

The activated SEL signals select one or more of the programmed switch elements in the switch circuitry 520 to output one or more bits of the input string on the CBUS as outputs Y<sub>0</sub>-Y<sub>L-1</sub> on signal lines 532<sub>0</sub>-532<sub>L-1</sub>.

**[0061]** One or more of the memory circuits 528 can be programmed with the same segment information. For one example, a first group of memory circuits are programmed to select a first group of switch circuits in switch circuitry 520 and a second group of memory circuits are programmed to select a second group of switch elements in switch circuitry 520. In this example, during a compare operation, SSDATA represents the first group of select circuits during a first compare operation such that the first group of switch circuits is selected to couple one or bits from the CBUS to corresponding bit locations of the comparand register; and, SSDATA represents the second group of select circuits during a second compare operation such that the second group of switch circuits is selected to couple one or bits from the CBUS to corresponding bit locations of the comparand register.

**[0062]** Each memory circuit/compare circuit pair (e.g., memory circuit 528<sub>0</sub> and compare circuit 526<sub>0</sub>) may be formed as a content addressable memory (CAM) cell. The CAM cell may be any type of CAM cell including be binary (capable of storing and comparing two logic states) or ternary (capable of storing and comparing three states - logic 1, logic 0 and don't care). The SEL signal lines 530, in this example, represent match lines of the CAM cells. In an alternative embodiment, select logic 524 is a programmable decoder.

2047754.01403

[0063] Figure 8 illustrates comparand register 541 that is one embodiment of comparand register 504. Comparand register 541 includes as many register elements  $540_0$ - $540_{L-1}$  as output signals  $Y_0$ - $Y_{L-1}$  from switch circuitry 520. Each register element 540 receives a clock signal WCLK from a clock generator (not shown) in order to synchronize the loading of  $Y_0$ - $Y_{L-1}$  into the register elements. Each register element 540 also receives an segment enable signal SEL that indicates whether the particular register element is enabled to load the output data  $Y_0$ - $Y_{L-1}$ .

[0064] Figure 9 illustrates cross-bar switch (XBAR) 600 that is one embodiment of switch circuitry 520. XBAR 600 includes an array of four rows and four columns of memory storage elements 602 each coupled to a pair of transistors 604 and 606. Any array size may be used (e.g., 72 rows by 72 columns, 144 rows by 144 columns, 288 rows by 288 columns, etc.). Memory storage elements 602 and transistor pair 604/606 each make up a programmable switch element. Transistors 604 and 606 are coupled in series between an output signal line 532 and an input signal line 533. Each memory storage element 602 is coupled to the gate of a transistor 604. Additionally, the source of each transistor 604 is coupled to the drain of transistors 606. SEL signal lines 530 are each coupled to the gate of a column of corresponding transistors 606, and the drains of each column of transistors 606 are coupled to a corresponding output signal line 532.

[0065] Each memory storage element 602 and transistor pair 604/606 is positioned at the intersection of a row and column. In this example, input signals IN each represent one bit of input string 508 from Figure 5, and output signals Y each represent one bit of a filtered comparand string provided to comparand register 504. Each memory storage element 602 can be programmed (e.g., by PDATA from program circuit 522 of Figure 6)

by one or more word lines and/or bit lines not shown so as not to obscure the figure.

During a compare operation, if a memory storage element is programmed to output a corresponding input bit IN as an output signal Y (e.g., programmed as a logic high state to enable transistor 604 to conduct), then the connection will be established if the corresponding SEL signal is also in a high logic state. In this manner, only those connections associated with particular segments are enabled at any one time.

[0066] XBAR 600 is a full cross-bar switch that enables any input IN to be connected to any output Y. In an alternative embodiment, only a portion of XBAR 600 may be needed such as when an input string is compacted in only one direction. For example, when compacting the input string from higher bit positions to lower bit positions in the output string, the corresponding circuitry of the cross-bar switch for translating lower bit positions to higher bit positions may be removed from a full cross-bar switch. Similarly, when compacting the input string from lower bit positions to higher significant bit positions in the output string, the corresponding circuitry of the cross-bar switch for translating higher bit positions to lower significant bits positions may be removed from a full cross-bar switch.

[0067] Figure 10 illustrates one example of an input string 700 that is filtered into a comparand register using, for example, the switch circuit of Figure 9. Input string 700 is one embodiment of input string 508 of Figure 5. Input string 700 has eight bits, bits 0-7 that are transmitted over the CBUS in two segments: segment 0 and segment 1. Segment 0 includes bits 0-3 and segment 1 includes bits 4-7. For this embodiment, the CBUS is L=4 bits wide; however, the CBUS could also be at least 8 bits wide. XBAR 600 can be programmed to perform the filter function on input string as shown in Figure 11. Memory

storage elements 602<sub>3,3</sub>-602<sub>3,0</sub> are programmed (e.g., by PDATA) as 0000, respectively; memory storage elements 602<sub>2,3</sub>-602<sub>2,0</sub> are programmed as 1000, respectively; memory storage elements 602<sub>1,3</sub>-601<sub>1,0</sub> are programmed as 0010, respectively; and, memory storage elements 602<sub>0,3</sub>-602<sub>0,0</sub> are programmed as 0101, respectively. Additionally, select logic 531 can be programmed to enable the appropriate SEL signals as shown in Figure 12 in which memory circuits 528<sub>3</sub> and 528<sub>2</sub> store logic one states and memory circuits 528<sub>1</sub> and 528<sub>0</sub> store logic zero states.

[0068] During segment 0 (e.g., during a first clock cycle), input signal line 533<sub>0</sub> carries bit 0 of input string 700 as IN<sub>0</sub>, input signal line 533<sub>1</sub> carries bit 1 of input string 700 as IN<sub>1</sub>, input signal line 533<sub>2</sub> carries bit 2 of input string 700 as IN<sub>1</sub>, and input signal line 533<sub>3</sub> carries bit 3 of input string 700 as IN<sub>1</sub>. During this segment, SSDATA is equal to a logic zero state to indicate that segment 0 of input string 700 is present on the CBUS. SSDATA is compared with the stored states in memory circuits 528 and each of SEL<sub>1</sub> and SEL<sub>0</sub> are set to high logic states indicating matches with data stored in memory circuits 528<sub>1</sub> and 528<sub>0</sub>, respectively, while SEL<sub>3</sub> and SEL<sub>2</sub> are set to logic zero states indicating mismatches with data stored in memory circuits 528<sub>3</sub> and 528<sub>2</sub>, respectively. Since SEL<sub>1</sub> and SEL<sub>0</sub> are in logic one states and memory storage elements 602<sub>1,1</sub> and 602<sub>0,0</sub> are programmed with logic one states, the input bits 1 and 0 of segment 0 of input string 700 are coupled to Y<sub>1</sub> and Y<sub>0</sub>, respectively, and are loaded into bits 1 and 0 of the comparand register. Since SEL<sub>3</sub> and SEL<sub>2</sub> are set to logic zero states, IN<sub>0</sub> and IN<sub>3</sub> are not transferred to Y<sub>2</sub> and Y<sub>3</sub>, respectively, because transistors 606 are off, and the corresponding locations in the comparand register are not updated. In this manner, bits 1

and 0 of input string 700 are filtered from the input string and loaded into bit positions 1 and 0 of the comparand register.

[0069] During segment 1 (e.g., during a second clock cycle), input signal line 533<sub>0</sub> carries bit 4 of input string 700 as IN<sub>0</sub>, input signal line 533<sub>1</sub> carries bit 5 of input string 700 as IN<sub>1</sub>, input signal line 533<sub>2</sub> carries bit 6 of input string 700 as IN<sub>2</sub>, and input signal line 533<sub>3</sub> carries bit 7 of input string 700 as IN<sub>3</sub>. During this segment, SSDATA is equal to a logic one state to indicate that segment 1 of input string 700 is present on the CBUS. SSDATA is compared with the stored states in memory circuits 528 and each of SEL<sub>3</sub> and SEL<sub>2</sub> are set to high logic states indicating matches with data stored in memory circuits 528<sub>3</sub> and 528<sub>2</sub>, respectively, while SEL<sub>1</sub> and SEL<sub>0</sub> are set to logic zero states indicating mismatches with data stored in memory circuits 528<sub>1</sub> and 528<sub>0</sub>, respectively. Since SEL<sub>3</sub> and SEL<sub>2</sub> are in logic one states and memory storage elements 602<sub>2,3</sub> and 602<sub>0,2</sub> are programmed with logic one states, the input bits 6 and 4 of segment 1 of input string 700 are coupled to Y<sub>3</sub> and Y<sub>2</sub>, respectively, and are loaded into bits 3 and 2 of the comparand register. Since SEL<sub>1</sub> and SEL<sub>0</sub> are set to logic zero states, IN<sub>1</sub> and IN<sub>0</sub> are not transferred to Y<sub>1</sub> and Y<sub>0</sub>, respectively, because transistors 606 are off, and the corresponding locations in the comparand register are not updated. In this manner, bits 4 and 6 of input string 700 are filtered from the input string and compacted into bit positions 2 and 3 of the comparand register.

[0070] Bits of the input string can also be transposed by XBAR 600 to different bit positions in the comparand register. For example, as shown in Figure 13, bits 5 and 7 in segment 1 of input string 700 can be transposed to bits 2 and 3, respectively, of the comparand register. XBAR 600 can be programmed as in Figure 14 such that memory

storage elements 602<sub>3,3</sub>-602<sub>3,0</sub> are programmed as 0100, respectively; memory storage elements 602<sub>2,3</sub>-602<sub>2,0</sub> are programmed as 0000, respectively; memory storage elements 602<sub>1,3</sub>-602<sub>1,0</sub> are programmed as 1010, respectively; and, memory storage elements 602<sub>0,3</sub>-602<sub>0,0</sub> are programmed as 0001, respectively. During segment 1, SEL<sub>3</sub> and SEL<sub>2</sub> are in logic one states and input bits 7 and 5 of segment 1 (presented as IN<sub>3</sub> and IN<sub>1</sub>, respectively) of input string 700 are coupled to Y<sub>2</sub> and Y<sub>3</sub>, respectively, and are loaded into bits 2 and 3 of the comparand register. In this manner, bits 7 and 5 of input string 700 are filtered from the input string and compacted and transposed into bit positions 3 and 2 of the comparand register.

[0071] With reference again to select logic 531 of Figure 7, the SSDATA indicates which segment of input data is present on the CBUS, and SSDATA is compared with data stored in each memory circuit 528 to determine which SEL signals are set to active high logic states. For the examples in Figure 12, each memory circuit 528 only needed to store a single bit because there were only two segments of input data on the CBUS. When there are more than two segments of input data for a particular input data string on the CBUS, the memory circuits 528 each store a sufficient number of bits of information to identify which segment of the input data should set the corresponding SEL signal to a high state. For example, if there are four segments of the input data, each memory circuit may store four bits (one for each segment) or two encoded bits to represent the four segments. Such an example is shown as select logic 831 of Figure 15.

[0072] Select logic 831 of Figure 15 includes L compare circuits 826<sub>0</sub>-826<sub>L-1</sub>, each coupled to a corresponding memory circuit 828<sub>0</sub>-828<sub>L-1</sub> that each have a pair of memory storage elements 820 and 822. The memory storage elements each store one bit of



SSDATA during a program or write operation (i.e., when R/W\* is in a logic low state).

One or more of the memory circuits 828 can be programmed with the same segment information representing up to four segments of the input data on the CBUS. For example, memory storage elements 820 are programmed by SSDATA<sub>1</sub> and memory storage elements 822 are programmed by SSDATA<sub>0</sub>. Each of the compare circuits compares the contents of the corresponding memory storage elements with SSDATA during a compare operation (i.e., when CMP is in a high logic state). That is, SSDATA<sub>1</sub> is compared with contents stored in each of memory storage elements 820 and SSDATA<sub>0</sub> is compared with contents stored in each of memory storage elements 822. The comparison results are reflected on the SEL signal lines 530<sub>0</sub>-530<sub>L-1</sub>.

[0073] In this example, SSDATA includes two bits such that up to four segments may be represented. The SSDATA signals may be generated, for example, by processor 310 of Figure 3 or provided by instruction decoder 510 of Figure 5 in response to a compare instruction for input data that is provided over multiple cycles. For example, if the compare instruction indicates that the input string has four segments, the SSDATA bits SSDATA<sub>1</sub> and SSDATA<sub>0</sub> may be logic states 00 to represent when the first segment of data is present on the CBUS, 01 to represent when the second segment of data is present on the CBUS, 10 to represent when the third segment of data is present on the CBUS, and 11 to represent when the fourth segment of data is present on the CBUS. For one embodiment, two-bit counter may be used to cycle through the four state in response to one or more control signals from the instruction decoder. Alternatively, a shift register may be used.

[0074] As with select logic 531 of Figure 7, each memory circuit/compare circuit pair (e.g., memory circuit 828 and compare circuit 826) may be formed as any type of CAM cells. For example, as shown in Figure 16, each compare circuit 826/memory circuit 828 pair includes two CAM cells 840 and 842. For this embodiment, compare circuit 826 includes two compare circuits 824 and 825. CAM cell 840 includes memory storage element 820 and compare circuit 824. CAM cell 842 includes memory storage element 822 and compare circuit 825. Compare circuit 824 compares  $SSDATA_1$  and the data stored in memory storage element 820 and provides the match result to the SEL signal line 530 which is a match signal line. Similarly, compare circuit 825 compares  $SSDATA_0$  and the data stored in memory storage element 822 and provides the match result to the SEL signal line 530.

[0075] With reference again to Figure 5, CAM array 502 may be a single CAM array that has Z rows of L CAM cells. The CAM array may be blocked out as X CAM array blocks each having R rows of L CAM cells, where Z equal R multiplied by X. In this embodiment, a first block has rows 0 to R-1, a second block has rows R to 2R-1, and so on, with the last (Xth) block having the last R rows. The CAM array can be blocked out to reduce the signal line length for bit lines, word lines and/or comparand lines in the arrays so as to reduce the impedance of the lines and increase the operating speed of the array as a whole. One embodiment of this arrangement is shown in Figure 17.

[0076] Figure 17 shows that CAM array 502 blocked out into X CAM array blocks  $902_1$ - $902_X$  each having a corresponding comparand register  $904_1$ - $904_X$  and a filter circuit  $906_1$ - $906_X$ . Each of the comparand registers 904 operate like comparand register 504 of Figure 5, and each of the filter circuits 906 operate like the filter circuit 506 of Figure 5.

Each of the filter circuits 906 receives FDATA and SSDATA and segments from the input string 508 such that each of the filter circuits is able to filter the input string for a corresponding comparand register. For this embodiment, each filter circuit 906 outputs the same signals to each of the corresponding comparand registers 904. Alternatively, one or more of the filter circuits is programmed differently than the other filter circuits to output different signals to the corresponding comparand register.

[0077] Each of the filter circuit 906 may operate concurrently (e.g., overlapping at least at one point in time) to simultaneously perform the filtering function and further increase overall throughput and performance of the device. Alternatively, one or more of the filter circuits 906 may perform its filtering of the input string before or after one of the other filter circuits 906.

[0078] As indicated above, each of the filter circuits 906 operate like the filter circuit 506 of Figure 5 and may include one of the various embodiments of the filter circuits described above. For one embodiment, each of the filter circuits 906 includes switch circuitry 520, select logic 524 and program circuit 522 as shown in Figure 6.

[0079] For an alternative embodiment, each of the CAM array blocks 902 may be a segment of a single CAM array 502 (or of a single CAM array block of a blocked out CAM array) having X segments each having Z rows of L CAM cells, where CAM array 502 has a total of Z rows of CAM cells.

[0080] For another embodiment, one or more of the filter circuits 906 shares the select logic 524 and/or the program circuit 522 from another one of the filter circuits. For other embodiments, a single filter circuit 906 and/or comparand register 904 may be shared by more than one block or segment 902, for example, by routing the output signal

lines from each comparand register to more than one block or segment 902. For still other embodiments, the program circuit 522 is not used.

[0081] In the foregoing specification, the invention is described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention as set forth in the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

1004754-01.1402